

[illegible]

SSSSSSSS	AAAAAA	TTTTTTTTTT	SSSSSSSS	SSSSSSSS	SSSSSSSS	333333	55555555
SSSSSSSS	AAAAAA	TTTTTTTTTT	SSSSSSSS	SSSSSSSS	SSSSSSSS	333333	55555555
SS	AA	TT	SS	SS	SS	33	55
SS	AA	TT	SS	SS	SS	33	55
SS	AA	TT	SS	SS	SS	33	555555
SS	AA	TT	SS	SS	SS	33	555555
SSSSSS	AA	TT	SSSSSS	SSSSSS	SSSSSS	33	55
SSSSSS	AA	TT	SSSSSS	SSSSSS	SSSSSS	33	55
SS	AAAAAAAAA	TT	SS	SS	SS	33	55
SS	AAAAAAAAA	TT	SS	SS	SS	33	55
SS	AA	TT	SS	SS	SS	33	55
SS	AA	TT	SS	SS	SS	33	55
SSSSSSSS	AA	TT	SSSSSSSS	SSSSSSSS	SSSSSSSS	333333	555555
SSSSSSSS	AA	TT	SSSSSSSS	SSSSSSSS	SSSSSSSS	333333	555555

• • • •

• • • •

• • • •

• • • •

```

LL               IIIIII               SSSSSSSS
LL               IIIIII               SSSSSSSS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LL               II                   SSSSSS
LL               II                   SSSSSS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS

```


(1)	77	DECLARATIONS
(1)	332	R/W PSECT
(1)	413	SATSSS35
(1)	462	CREPRC TESTS
(1)	764	GETJPI TESTS
(2)	969	ROUTINES
(2)	970	REG_SAVE
(2)	991	REG_CHECK
(2)	1033	PRINT_FAIL
(2)	1080	MODE_ID
(2)	1102	CRE_CHECK
(2)	1143	JPI_CHECK


```
0000 1 .TITLE SATSSS35 - SATS SYSTEM SERVICE TESTS (SUCC S.C.)
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 *****
0000 27
0000 28
0000 29 ++
0000 30 FACILITY: SATS SYSTEM SERVICE TESTS
0000 31
0000 32 ABSTRACT: The SATSSS35 module tests the execution of the following
0000 33 VMS system services:
0000 34
0000 35 $CREPRC
0000 36 $GETJPI
0000 37
0000 38 ENVIRONMENT: User mode image.
0000 39 Needs CMKRN privilege and dynamically acquires other
0000 40 privileges, as needed.
0000 41
0000 42 AUTHOR: Larry D. Jones, CREATION DATE: JULY, 1979
0000 43
0000 44 MODIFIED BY:
0000 45
0000 46 V03-002 LDJ0006 Larry D. Jones, 23-Mar-1982
0000 47 Made the quota list be absolute minimum to test the
0000 48 SYSBOOT minimum values.
0000 49
0000 50 V03-001 RNP0005 Robert N. Perron, 23-Mar-1982
0000 51 Removed EXCVEC and FINALEXC from the JPI_GOOD list.
0000 52
0000 53 V02-006 RNP0004 Robert N. Perron, 09-Dec-1981
0000 54 Removed ASTEN from the JPI_GOOD list.
0000 55
0000 56 V02-005 RNP0003 Robert N. Perron, 02-Oct-1981
0000 57 Removed ASTACT from the JPI_GOOD list.
```


0000	58	:	
0000	59	:	
0000	60	:	V02-004 LDJ0002 Larry D. Jones, 06-Sep-1981
0000	61	:	Fixed GETJPI P1 reference to CTL\$AQ_EXCVEC and CTL\$AL_FINALEXC.
0000	62	:	
0000	63	:	V02-003 RNP0002 Robert N. Perron, 01-Jun-1981
0000	64	:	To eliminate dependence on the SYSTEST account privileges
0000	65	:	being a specific list, privileges are now set to a fixed list
0000	66	:	before the GETJPI tests are started.
0000	67	:	
0000	68	:	V02-002 RNP0001 Robert N. Perron, 09-Apr-1981
0000	69	:	Fixed problem of STS field changing due to Swapper activity.
0000	70	:	Prevent failure when privileges are added to the SYSTEST
0000	71	:	account. Cleaned up some format problems.
0000	72	:	
0000	73	:	V02-001 LDJ0001 Larry D. Jones, 17-Sep-1980
0000	74	:	Modified to conform to new build command procedures.
0000	75	:	


```
0000 77      .SBTTL  DECLARATIONS
0000 78      :
0000 79      : MACRO LIBRARY CALLS
0000 80      :
0000 81      $ACCDDEF      ; account record offset definitions
0000 82      $DIBDEF      ; device info block definitions
0000 83      $JPIDEF      ; JPI offset definitions
0000 84      $PCBDEF      ; process control block definitions
0000 85      $PHDDEF      ; Process header definitions
0000 86      $PQLDEF      :
0000 87      $PRVDEF      ; privilege definitions
0000 88      $SHRDEF      ; shared message definitions
0000 89      $SFDEF       ; stack frame definitions
0000 90      $STSDEF      ; STS definitions
0000 91      $UETPDEF     ; UETP message definitions
0000 92      :
0000 93      :
00000001 0000 94      SUCCESS      = 1      ; success
00000002 0000 95      ERROR       = 2      ; error
0000 96      :
0000 97      : SHR message definitions
0000 98      :
00740000 0000 99      UETP = 116@STSV_FAC_NO      ;define the UETP facility code
0000 100     :
00741038 0000 101     UETPS_BEGIN = UETP!SHRS_BEGIN ;define the UETP messages
00741130 0000 102     UETPS_TEXT  = UETP!SHRS_TEXT
007410E0 0000 103     UETPS_ABEND = UETP!SHRS_ABEND
00741080 0000 104     UETPS_ENDED = UETP!SHRS_ENDED
0000 105     :
0000 106     : Mask of bits for the STS field in a $CREPRC system service as they are
0000 107     : returned from a $GETJPI system service.
0000 108     :
0000 109     JPI_STS_MASK = <<1@PCBSV_NETWORK>!<1@PCBSV_SSFEXCU>!<1@PCBSV_SSRWAIT>!-
0000 110     <1@PCBSV_BATCH>!<1@PCBSV_NOACNT>!<1@PCBSV_HIBER>!-
0038C600 0000 111     <1@PCBSV_LOGIN>>
0000 112     :
0000 113     : The opposite of JPI_STS_MASK
0000 114     :
0000 115     JPI_STS_NMASK = ^CJPI_STS_MASK
0000 116     :
0000 117     : Mask of bits for the Privilege field as they are returned from a $GETJPI
0000 118     :
0000 119     :
0000 120     JPI_PRV_MASK = <<1@PRVSV_CMEXEC>!<1@PRVSV_CMKRNL>!<1@PRVSV_DETACH>!-
0000 121     <1@PRVSV_DIAGNOSE>!<1@PRVSV_GROUP>!<1@PRVSV_GRPNAM>!-
0000 122     <1@PRVSV_LOG_IO>!<1@PRVSV_NETMBX>!<1@PRVSV_NOACNT>!-
0000 123     <1@PRVSV_PHY_IO>!<1@PRVSV_PRMCB>!<1@PRVSV_PRMMBX>!-
0000 124     <1@PRVSV_PSWAPM>!<1@PRVSV_SETPRI>!<1@PRVSV_SYSNAM>!-
0000 125     <1@PRVSV_SYSPRV>!<1@PRVSV_TPMBX>!<1@PRVSV_VOLPRO>>
0000 126     :
0000 127     : The compliment of JPI_PRV_MASK
0000 128     :
0000 129     JPI_PRV_NMASK = ^CJPI_PRV_MASK
0000 130     :
0000 131     : MACROS
0000 132     :
0000 133     .MACRO JPI,NAME,SIZE
```


SATSSS35
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00 Page 4
DECLARATIONS 5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1 (1)

```
0000 134      .WORD  SIZE
0000 135      .WORD  JPI$ 'NAME'
0000 136      .ADDRESS NAME
0000 137      .ADDRESS NAME'L
0000 138      .SAVE  PSECT
0000 139      .PSECT ITEM_LIST
0000 140 NAME:
0000 141      .BLKB  SIZE
0000 142 NAME'L:
0000 143      .WORD  0
0000 144      .RESTORE PSECT
0000 145      .ENDM JPI
```


SATSSS35
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
DECLARATIONS 5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 5
(1)

```
00000000 147 .PSECT ITEM_LIST, RD, WRT, NOEXE, LONG ; psect to store JPI results in
00000000 148 .PSECT RODATA, RD, NOWRT, NOEXE, LONG
0000 149 ;
0000 150 TEST_MOD_NAME:
35 33 53 53 53 54 41 53 00' 0000 151 .ASCIC /SATSSS35/ ; needed for SATSMS message
08 0000
0009 152 TEST_MOD_NAME_D:
53 53 53 54 41 53 00000011'010E0000' 0009 153 .ASCID /SATSSS35/ ; module name
35 33 0017
0019 154 TEST_MOD_BEGIN: ; start end and fail messages
6E 75 67 65 62 00' 0019 155 .ASCIC /begun/
05 0019
001F 156 TEST_MOD_SUCC:
6C 75 66 73 73 65 63 63 75 73 00' 001F 157 .ASCIC /successful/
0A 001F
002A 158 TEST_MOD_FAIL:
64 65 6C 69 61 66 00' 002A 159 .ASCIC /failed/
06 002A
0031 160 CS1: ; failure messages
21 20 74 73 65 54 00000039'010E0000' 0031 161 .ASCID \Test !AC service name !AC step !UL failed.\
6E 20 65 63 69 76 72 65 73 20 43 41 003F
70 65 74 73 20 43 41 21 20 65 6D 61 004B
2E 64 65 6C 69 61 66 20 4C 55 21 20 0057
0063 162 CS2:
74 63 65 70 78 45 0000006B'010E0000' 0063 163 .ASCID \Expected !AS = !XL received !AS = !XL\
4C 58 21 20 3D 20 53 41 21 20 64 65 0071
41 21 20 64 65 76 69 65 63 65 72 20 007D
4C 58 21 20 3D 20 53 0089
0090 164 CS3:
74 63 65 70 78 45 00000098'010E0000' 0090 165 .ASCID \Expected !AS!UB = !XL received !AS!UB = !XL\
20 3D 20 42 55 21 53 41 21 20 64 65 009E
64 65 76 69 65 63 65 72 20 4C 58 21 00AA
58 21 20 3D 20 42 55 21 53 41 21 20 00B6
4C 00C2
00C3 166 CS5:
77 20 65 64 6F 4D 000000CB'010E0000' 00C3 167 .ASCID \Mode was !AS.\
2E 53 41 21 20 73 61 00D1
00D8 168 EXP:
73 75 74 61 74 73 000000E0'010E0000' 00D8 169 .ASCID \status\
00E6 170 AST_PARAM:
61 70 20 54 53 41 000000EE'010E0000' 00E6 171 .ASCID \AST param.\
2E 6D 61 72 00F4
00F8 172 BP:
70 20 65 73 61 62 00000100'010E0000' 00F8 173 .ASCID \base pri.\
2E 69 72 0106
0109 174 PNS:
73 65 63 6F 72 50 00000111'010E0000' 0109 175 .ASCID \Process name was not set correctly.\
6E 20 73 61 77 20 65 6D 61 6E 20 73 0117
65 72 72 6F 63 20 74 65 73 20 74 6F 0123
2E 79 6C 74 63 012F
0134 176 STSFLGS:
47 4C 46 53 54 53 0000013C'010E0000' 0134 177 .ASCID \STSFLG's\
73 27 0142
0144 178 UIC_MSG:
43 49 55 0000014C'010E0000' 0144 179 .ASCID \UIC\
014F 180 EFC_NAME:
46 53 53 54 41 53 00000157'010E0000' 014F 181 .ASCID \SATSSF06_DET\
```



```
54 45 44 5F 36 30 015D
44 49 50 0000016B'010E0000' 0163 182 PID_STR:
43 52 50 45 52 43 00' 0163 183 .ASCID \PID\
06 016E 184 CREPRC:
016E 185 .ASCIC \CREPRC\
0175 186 GETJPI:
49 50 4A 54 45 47 00' 0175 187 .ASCIC \GETJPI\
06 0175
017C 188 UM: ; mode messages
72 65 73 75 00000184'010E0000' 017C 189 .ASCID \user\
0188 190 MBNAM:
58 42 4D 35 33 53 00000190'010E0000' 0188 191 .ASCID \S35MBX\
0196 192 PRVMASK:
00000000 1070BF EF 0196 193 .QUAD JPI_PRV_MASK ; used for setting privileges to
019E 194 ; known value
019E 195 NPRVMASK:
00000000 EF8F4010 019E 196 .QUAD JPI_PRV_NMASK ; used for clearing any extra
01A6 197 ; privileges
01A6 198 MSGVEC: ; PUTMSG message vector
00000003 01A6 199 .LONG 3
00741130 01AA 200 .LONG UETPS_TEXT
00000001 01AE 201 .LONG 1
00000217' 01B2 202 .ADDRESS MESSAGEL
01B6 203 QUOTA_LIST:
01 01B6 204 .BYTE PQLS_ASTLM ; minimum quota list
00000001 01B7 205 .LONG 1
02 01BB 206 .BYTE PQLS_BIOLM
00000001 01BC 207 .LONG 1
03 01C0 208 .BYTE PQLS_BYTLM
00000001 01C1 209 .LONG 1
04 01C5 210 .BYTE PQLS_CPULM
00000000 01C6 211 .LONG 0
05 01CA 212 .BYTE PQLS_DIOLM
00000001 01CB 213 .LONG 1
06 01CF 214 .BYTE PQLS_FILLM
00000001 01D0 215 .LONG 1
07 01D4 216 .BYTE PQLS_PGFLQUOTA
00000001 01D5 217 .LONG 1
08 01D9 218 .BYTE PQLS_PRCLM
00000000 01DA 219 .LONG 0
09 01DE 220 .BYTE PQLS_TQELM
00000000 01DF 221 .LONG 0
0B 01E3 222 .BYTE PQLS_WSDEFAULT
00000001 01E4 223 .LONG 1
0A 01E8 224 .BYTE PQLS_WSQUOTA
00000001 01E9 225 .LONG 1
00 01ED 226 .BYTE PQLS_LISTEND
```



```
01EE 228 GET_LIST: ; GETJPI list of items and results
01EE 229 JPI ACCOUNT,8
01FA 230 SHORT_LIST:
01FA 231 JPI CPULIM,4
0206 232 JPI CURPRIV,8
0212 233 JPI GRP,4
021E 234 JPI IMAGPRIV,8
022A 235 JPI MEM,4
0236 236 JPI PRCLM,4
0242 237 JPI TQLM,4
024E 238 JPI UIC,4
025A 239 JPI USERNAME,12
00000044 0266 240 JPI LIST_SIZE=<USERNAME+2>-ACCOUNT
0000003A 0266 241 JPI LIST_SIZE1=<USERNAME+2>-CPULIM
0266 242 DIRTY: ; GETJPI entrys which will vary
0266 243 JPI APTCNT,4
0272 244 JPI ASTACT,4
027E 245 JPI ASTEN,4
028A 246 JPI ASTCNT,4
0296 247 JPI ASTLM,4
02A2 248 JPI AUTHPRIV,8
02AE 249 JPI BIOCNT,4
02BA 250 JPI BIOLM,4
02C6 251 JPI BUFIO,4
02D2 252 JPI BYTCNT,4
02DE 253 JPI BYTLM,4
02EA 254 JPI CPUTIM,4
02F6 255 JPI DFPFC,4
0302 256 JPI DFWSCNT,4
030E 257 JPI DIOCNT,4
031A 258 JPI DIOLM,4
0326 259 JPI DIRIO,4
0332 260 JPI EFCS,4
033E 261 JPI EFCU,4
034A 262 JPI EFWM,4
0356 263 JPI EXCVEC,4
0362 264 JPI FINALEXC,4
036E 265 JPI FILCNT,4
037A 266 JPI FILLM,4
0386 267 JPI FREPOVA,4
0392 268 JPI FREPIVA,4
039E 269 JPI GPGCNT,4
03AA 270 JPI IMAGNAME,128
03B6 271 JPI LOGINTIM,4
03C2 272 JPI OWNER,4
03CE 273 JPI PAGEFLTS,4
03DA 274 JPI PGFLQUOTA,4
03E6 275 JPI PID,4
03F2 276 JPI PPGCNT,4
03FE 277 JPI PRCCNT,4
040A 278 JPI PRCNAM,15
0416 279 JPI PROCPRIV,8
0422 280 JPI PRI,4
042E 281 JPI PRI8,4
043A 282 JPI STATE,4
0446 283 JPI STS,4
0452 284 JPI TMBU,4
```


SATSSS35
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.)^{K 8} 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
DECLARATIONS 5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 8
(1)

	045E	285	JPI TQCNT,4	
	046A	286	JPI VOLUMES,4	
	0476	287	JPI VIRTPEAK,4	
	0482	288	JPI WSAUTH,4	
	048E	289	JPI WSQUOTA,4	
	049A	290	JPI WSPEAK,4	
	04A6	291	JPI WSSIZE,4	
00000000	04B2	292	.LONG 0	; list terminator


```

20 54 53 45 54 53 59 53 0008 04B6 294 JPI_GOOD: ; expected GETJPI results
                                04B6 295 ; Item name      buffer offset
                                04B6 296 ; ACCOUNT        00
                                04BE 297 ; ACCOUNTL       08
                                04C0 298 JPI_GOOD-SHRT:
                                04C0 299 ; LONG          0
                                04C4 300 ; WORD          4
                                04C6 301 ; QUAD          JPI_PRV_MASK
                                04CE 302 ; WORD          8
                                04D0 303 ; LONG          1
                                04D4 304 ; WORD          2
                                04D6 305 ; QUAD          0
                                04DE 306 ; WORD          8
                                04E0 307 ; LONG          7
                                04E4 308 ; WORD          2
                                04E6 309 ; LONG          8
                                04EA 310 ; WORD          2
                                04EC 311 ; LONG          ^X14
                                04F0 312 ; WORD          2
                                04F2 313 ; LONG          ^X10007
                                04F6 314 ; WORD          4
                                04F8 315 ; ASCII /SYSTEST /
                                0504 316 ; WORD          ^XC
20 20 20 20 20 54 53 45 54 53 59 53 000C 0504 317 IN: ;
                                0506 318 ; ASCII /SYS$INPUT/
4E 49 24 53 59 53 0000050E'010E0000' 0506 319 OUT:
                                0514 320 ; ASCII /SYS$OUTPUT/
55 4F 24 53 59 53 0000051F'010E0000' 0517 321 ERR:
                                0525 322 ; ASCII /SYS$ERROR/
52 45 24 53 59 53 00000531'010E0000' 0529 323 IMAGE_NAME:
                                0537 324 ; ASCII /SATSUT01.EXE/
54 55 53 54 41 53 00000542'010E0000' 053A 325 PROC_NAME:
                                0548 326 ; ASCII /SATSUT35/
54 55 53 54 41 53 00000556'010E0000' 054E 327 ; BLKB      7
                                055C 328 PROC_UIC:
                                055E 329 ; LONG      ^X10007
                                0565 ; process UIC
                                0565
```



```
0569 331 ;
0569 332 .SBTTL R/W PSECT
00000000 333 .PSECT RWDATA,RD,WRT,NOEXE, LONG
0000 334 ;
0000 335 ;PID:
00000000 336 .LONG 0 ; PID for this process
00000000 337 CURRENT_TC: ; ptr to current test case
00000000 338 .LONG 0 ; put it on a long word boundry
0008 339 .ALIGN LONG
0008 340 REG_SAVE_AREA:
0008 341 .BLKL 15 ; register save area
0044 342 MOD_MSG_CODE:
007480D9 343 .LONG UETPS$SATSMS ; test module message code for putmsg
0048 344 TMN_ADDR:
00000000' 0048 345 .ADDRESS TEST_MOD_NAME
004C 346 TMD_ADDR:
00000019' 004C 347 .ADDRESS TEST_MOD_BEGIN
0050 348 PRVPRT:
00 0050 349 .BYTE 0 ; protection return byte for SETPRT
00000000 00000000 0051 350 PRIVMASK: ; priv. mask
0059 351 .QUAD 0
00000000 0059 352 CHM_CONT: ; change mode continue address
005D 353 .LONG 0
00000065 005D 354 RETADR: ; returned address's from SETPRT
0065 355 .BLKL 2
00000000 0065 356 STATUS:
0069 357 .LONG 0
00000000 0069 358 MODE: ; current mode string pointer
006D 359 .LONG 0
74 73 69 67 65 72 00000075'010E0000' 006D 360 REG:
52 20 72 65 007B 361 .ASCID \register R\
007F 362 REGNUM:
00000000 007F 363 .LONG 0 ; register number
0083 364 MSGL:
00000050 0083 365 .LONG 80 ; buffer desc.
00000133' 0087 366 .ADDRESS BUF
008B 367 CRE:
008B 368 $CREPRC PID1,0,0,0,0,0,QUOTA_LIST,-
008B 369 0,0,0,0,0 ; CREPRC parameter list
00C3 370 GET: ; GETJPI parameter list's
00C3 371 $GETJPI EFN=1,PIDADR=PID1,PRCNAM=TEST_MOD_NAME_D,ITMLST=GET_LIST
00E3 372 GET1:
00E3 373 $GETJPI ITMLST=GET_LIST
0103 374 ITEM_LIST:
00000133 0103 375 .BLKL 12
0133 376 BUF:
00000183 0133 377 .BLKB 80
0183 378 ML:
00000000 0183 379 .LONG 0 ; desc. for BUF_CHECK routine
00000193' 0187 380 .ADDRESS GETBUF+8
018B 381 GETBUF:
00000084 018B 382 .LONG 132
00000193' 018F 383 .ADDRESS +4
00000217 0193 384 .BLKB 132
0217 385 MESSAGEL:
00000000 0217 386 .LONG 0 ; message desc.
```


00000133'	021B	387	.ADDRESS BUF	
	021F	388	SERV_NAME:	
00000000	021F	389	.LONG	0
	0223	390	MSGVEC1:	
00000003	0223	391	.LONG	3
00741130	0227	392	.LONG	UETPS_TEXT
00000001	022B	393	.LONG	1
00000000	022F	394	.LONG	0
	0233	395	IOSTAT:	
00000000 00000000	0233	396	.QUAD	0
	023B	397	PID1:	
00000000	023B	398	.LONG	0
	023F	399	MBCHAN:	
0000	023F	400	.WORD	0
	0241	401	MBXUN:	
0000	0241	402	.WORD	0
	0243	403	IOSTATUS:	
0000024B	0243	404	.BLKL	2
	024B	405	MBUF:	
000002AF	024B	406	.BLKB	100
	02AF	407	TEST_PID:	
00000000	02AF	408	.LONG	0
	02B3	409	PRIVS:	
00000000 00000000	02B3	410	.QUAD	0

; service name pointer
; PUTMSG message vector
; IO status block
; PID storage location
; MBX channel location
; MBX unit number
; MBX read IO status block
; MBX read buffer
; GETJPI parameter
; privilege mask


```
00000000 412 .PSECT SATSSS35, RD, WRT, EXE, LONG
0000 413 .SBTTL SATSSS35
0000 414 :++
0000 415 : FUNCTIONAL DESCRIPTION:
0000 416 :
0000 417 : After performing some initial housekeeping, such as
0000 418 : printing the module begin message and acquiring needed privileges,
0000 419 : the system services are tested in each of their normal conditions.
0000 420 : Detected failures are identified and an error message is printed
0000 421 : on the terminal. Upon completion of the test a success or fail
0000 422 : message is printed on the terminal.
0000 423 :
0000 424 : CALLING SEQUENCE:
0000 425 :
0000 426 : $ RUN SATSSS35 ... (DCL COMMAND)
0000 427 :
0000 428 : INPUT PARAMETERS:
0000 429 :
0000 430 : none
0000 431 :
0000 432 : IMPLICIT INPUTS:
0000 433 :
0000 434 : none
0000 435 :
0000 436 : OUTPUT PARAMETERS:
0000 437 :
0000 438 : none
0000 439 :
0000 440 : IMPLICIT OUTPUTS:
0000 441 :
0000 442 : Messages to SYS$OUTPUT are the only output from SATSSS35.
0000 443 : They are of the form:
0000 444 :
0000 445 : %UETP-S-SATSMS, TEST MODULE SATSSS35 BEGUN ... (BEGIN MSG)
0000 446 : %UETP-S-SATSMS, TEST MODULE SATSSS35 SUCCESSFUL ... (END MSG)
0000 447 : %UETP-E-SATSMS, TEST MODULE SATSSS35 FAILED ... (END MSG)
0000 448 : %UETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
0000 449 :
0000 450 : COMPLETION CODES:
0000 451 :
0000 452 : The SATSSS35 routine terminates with a $EXIT to the
0000 453 : operating system with a status code defined by UETPS_SATSMS.
0000 454 :
0000 455 : SIDE EFFECTS:
0000 456 :
0000 457 : none
0000 458 :
0000 459 : --
0000 460 :
0000 461 : TEST_START SATSSS35 ; let the test begin
```



```
0000 0000 .ENTRY SATSSS35,0
0004'CF 00 D4 0002 CLRL W^CURRENT_TC
00 DD 0006 PUSHL #0
0000'CF 00 DF 0008 PUSHAL W^TPID
00000000'GF 02 FB 000C CALLS #2,G^SYSS$WAKE
00000000'GF 00 FB 0013 CALLS #0,G^SYSS$HIBER
0009'CF 01 7F 001A PUSHAQ W^TEST MOD NAME_D
00000000'GF 01 FB 001E CALLS #1,G^SYSS$SETPRN
004C'CF 001F'CF 30 0025 BSBW W^MOD MSG PRINT
0044'CF 03 00 DE 0028 MOVAL W^TEST MOD SUCC,W^TMD_ADDR
00 00 FO 002F INSV #SUCCESS,#0,#3,W^MOD_MSG_CODE
0BF3'CF 01 FB 0036 PUSHL #0
003D 0038 CALLS #1,W^REG_SAVE
003D 462 STP0: .SBTTL CREPRC TESTS
003D 463 :+
003D 464 :
003D 465 : $CREPRC tests
003D 466 :
003D 467 : test the minimum quota all defaults subprocess with _S
003D 468 :
003D 469 :-
0069'CF 017C'CF DE 003D 470 MOVAL W^UM,W^MODE ; set the mode
021F'CF 016E'CF DE 0044 471 MOVAL W^CREPRC,W^SERV_NAME ; set the service name
004B 472 $CREMBX_S CHAN=W^MBCHAN,-
004B 473 LOGNAM=W^MBNAM ; make something to listen with
0062 474 $GETCHN_S CHAN=W^MBCHAN,-
0062 475 PRIBUF=W^GETBUF ; get the unit number
0241'CF 019F'CF B0 0078 476 MOVW W^GETBUF+8+DIB$W_UNIT,W^MBXUN ; and save it
007F 477 $CREPRC_S QUOTA=W^QUOTA_LIST,-
007F 478 MBXUNT=W^MBXUN ; create a subprocess with _S
00A5 479 FAIL_CHECK SSS_NORMAL ; check for success
00000000'8F DD 00A5
0BFD'CF 01 FB 00AB
56 00000000'8F D0 00B0 480 MOVL #RMS$_FNF,R6 ; set exit status code
00 57 D4 00B7 481 CLRL R7 ; disable PID checking this time
0D30'CF 00 FB 00B9 482 CALLS #0,W^CRE_CHECK ; check the process exit code
00BE 483 :+
00BE 484 :
00BE 485 : test the PIDADR parameter with _G
00BE 486 :
00BE 487 :-
00BE 488 NEXT_TEST
00BE
0004'CF 01 D0 00BE STP1:
00 DD 00C3
0BF3'CF 01 FB 00C5
00B7'CF 0241'CF B0 00CA 489 MOVW W^MBXUN,W^CRE+CREPRC$_MBXUNT ; set the MBX unit number
00D1 490 $CREPRC G W^CRE ; try _G and PIDADR param.
00DA 491 FAIL_CHECK SSS_NORMAL ; check for success
00DA
00000000'8F DD 00DA
0BFD'CF 01 FB 00E0
00 57 D6 00E5 492 INCL R7 ; enable PID checking
0D30'CF 00 FB 00E7 493 CALLS #0,W^CRE_CHECK ; check the process exit code
56 00000000'8F D0 00EC 494 MOVL #SS$_NORMAL,R6 ; set expected status return
```



```
00F3 495 :+
00F3 496 :
00F3 497 : test the IMAGE param. with _S
00F3 498 :
00F3 499 :-
00F3 500 NEXT_TEST
00F3
00F3 STP2:
0004'CF 02 DO 00F3 MOVL #2,W^CURRENT_TC
00F3 DD 00F8 PUSHL #0
0BF3'CF 01 FB 00FA CALLS #1,W^REG_SAVE
00FF 501 $CREPRC_S QUOTA=W^QUOTA_LIST,-
00FF 502 IMAGE=W^IMAGE_NAME,-
00FF 503 MBXUNT=W^MBXUN,-
00FF 504 PIDADR=W^PID1 ; try _S with IMAGE param.
0129 505 FAIL_CHECK SSS_NORMAL ; check success
00000000'8F DD 0129 PUSHL #SS$ NORMAL
0BFD'CF 01 FB 012F CALLS #1,W^REG_CHECK
0D30'CF 00 FB 0134 506 $WAKE_S PIDADR = W^PID1 ; cause process termination
0141 507 CALLS #0,W^CRE_CHECK ; check the process exit code
0146 508 :+
0146 509 :
0146 510 : test the INPUT param. with _G
0146 511 :
0146 512 :-
0146 513 NEXT_TEST
0146
0146 STP3:
0004'CF 03 DO 0146 MOVL #3,W^CURRENT_TC
00F3'CF 00 DD 0148 PUSHL #0
0BFD'CF 01 FB 014D CALLS #1,W^REG_SAVE
0093'CF 053A'CF DE 0152 514 MOVAL W^IMAGE_NAME,W^CRE+CREPRC$ IMAGE ; set image name
008F'CF 023B'CF DE 0159 515 MOVAL W^PID1,W^CRE+CREPRC$ PIDADR ; set the PID save address
0097'CF 0506'CF DE 0160 516 MOVAL W^IN,W^CRE+CREPRC$ INPUT ; set the INPUT param.
0167 517 $CREPRC G W^CRE ; try it
0170 518 FAIL_CHECK SSS_NORMAL ; check success
00000000'8F DD 0170 PUSHL #SS$ NORMAL
0BFD'CF 01 FB 0176 CALLS #1,W^REG_CHECK
0D30'CF 00 FB 0178 519 $WAKE_S PIDADR = W^PID1 ; cause process termination
018D 520 CALLS #0,W^CRE_CHECK ; check the process exit code
018D 521 :+
018D 522 :
018D 523 : test the OUTPUT param. with _S
018D 524 :
018D 525 :-
018D 526 NEXT_TEST
018D
018D STP4:
0004'CF 04 DO 018D MOVL #4,W^CURRENT_TC
00F3'CF 00 DD 0192 PUSHL #0
0BFD'CF 01 FB 0194 CALLS #1,W^REG_SAVE
0199 527 $CREPRC_S PIDADR=W^PID1,-
0199 528 IMAGE=W^IMAGE_NAME,-
0199 529 INPUT=W^IN,-
0199 530 OUTPUT=W^OUT,-
0199 531 MBXUNT=W^MBXUN,-
0199 532 QUOTA=W^QUOTA_LIST ; try _S with OUT param.
```



```
00000000'8F DD 01C7 533 FAIL_CHECK SSS_NORMAL ; chec success
OBFD'CF 01 FB 01C7 PUSHL #SS$ NORMAL
0D30'CF 00 FB 01CD CALLS #1,W^REG_CHECK
01D2 534 $WAKE_S PIDADR = W^PID1 ; cause process termination
01DF 535 CALLS #0,W^CRE_CHECK ; check process exit code
01E4 536 :+
01E4 537 : test ERROR param. with _G
01E4 538 :-
01E4 539 :
01E4 540 :
01E4 541 NEXT_TEST
01E4 STP5:
0004'CF 05 DO 01E4 MOVL #5,W^CURRENT_TC
0000'CF 00 DD 01E9 PUSHL #0
0BF3'CF 01 FB 01EB CALLS #1,W^REG_SAVE
009B'CF 0517'CF DE 01F0 542 MOVAL W^OUT,W^CRE+CREPRCS_OUTPUT ; set the output param.
009F'CF 0529'CF DE 01F7 543 MOVAL W^ERR,W^CRE+CREPRCS_ERROR ; set the error output param
01FE 544 $CREPRC G W^CRE ; try _G with ERROR param
0207 545 FAIL_CHECK SSS_NORMAL ; check for success
00000000'8F DD 0207 PUSHL #SS$ NORMAL
OBFD'CF 01 FB 020D CALLS #1,W^REG_CHECK
0D30'CF 00 FB 0212 546 $WAKE_S PIDADR = W^PID1 ; cause process termination
021F 547 CALLS #0,W^CRE_CHECK ; check process exit code
0224 548 :+
0224 549 : test PRVADR param with _S
0224 550 :-
0224 551 :
0224 552 :
0224 553 NEXT_TEST
0224 STP6:
0004'CF 06 DO 0224 MOVL #6,W^CURRENT_TC
0000'CF 00 DD 0229 PUSHL #0
0BF3'CF 01 FB 022B CALLS #1,W^REG_SAVE
0230 554 $CREPRC_S PIDADR=W^PID1,-
0230 555 IMAGE=W^IMAGE_NAME,-
0230 556 INPUT=W^IN,-
0230 557 OUTPUT=W^OUT,-
0230 558 ERROR=W^ERR,-
0230 559 PRVADR=W^PRIVS,-
0230 560 MBXUNT=W^MBXUN,-
0230 561 QUOTA=W^QUOTA_LIST ; try _S with PRVADR param
0262 562 FAIL_CHECK SSS_NORMAL ; check success
00000000'8F DD 0262 PUSHL #SS$ NORMAL
OBFD'CF 01 FB 0268 CALLS #1,W^REG_CHECK
0D30'CF 00 FB 026D 563 $WAKE_S PIDADR = W^PID1 ; cause process termination
027A 564 CALLS #0,W^CRE_CHECK ; check image exit status
027F 565 :+
027F 566 : test PRCNAM param with _G
027F 567 :-
027F 568 :
027F 569 :
027F 570 NEXT_TEST
027F STP7:
0004'CF 07 DO 027F MOVL #7,W^CURRENT_TC
```



```
00          00          DD 0284          PUSHL #0
OBF3'CF 01          FB 0286          CALLS #1,W^REG_SAVE
009F'CF 0529'CF  DE 028B 571          MOVAL W^ERR,W^CRE+CREPRC$ ERROR ; set the ERROR param.
00A3'CF 02B3'CF  DE 0292 572          MOVAL W^PRIVS,W^CRE+CREPRC$ PRIVADR ; set the PRIVADR param.
00AB'CF 054E'CF  DE 0299 573          MOVAL W^PROC_NAME,W^CRE+CREPRC$_PRCNAM ; set the process name
                                02A0 574          $CREPRC G W^CRE ; try G with a PRCNAM
                                02A9 575          FAIL_CHECK SSS_NORMAL ; check success
                                DD 02A9          PUSHL #SS$ NORMAL
                                FB 02AF          CALLS #1,W^REG_CHECK
021F'CF 0175'CF  DE 02B4 576          MOVAL W^GETJPI,W^SERV_NAME ; set service name
                                02BB 577          $GETJPI_S PIDADR = W^PID1,-
                                02BB 578          ITMLST = W^GET_LIST ; get the process name
                                02D2 579          FAIL_CHECK SSS_NORMAL ; check success
                                DD 02D2          PUSHL #SS$ NORMAL
                                FB 02D8          CALLS #1,W^REG_CHECK
0556'CF 01A2'CF  DE 02DD 580          MOVAL W^CREPRC,W^SERV_NAME ; set service name
                                29 02E4 581          CMPC3 #15,W^PRCNAM,W^PROC_NAME+8 ; correct process name?
                                13 02EC 582          BEQL 10$ ; br if OK
                                DF 02EE 583          PUSHAL W^PNS ; push string variable
                                FB 02F2 584          CALLS #1,W^PRINT_FAIL ; print the failure
                                02F7 585 10$:
                                02F7 586          $WAKE_S PIDADR = W^PID1 ; cause process termination
                                FB 0304 587          CALLS #0,W^CRE_CHECK ; check image exit status
                                0309 588 ;+
                                0309 589 ;
                                0309 590 ; test BASPRI with _S and a lower priority
                                0309 591 ;
                                0309 592 ; -
                                0309 593
                                NEXT_TEST
                                STP8:
                                DD 0309          MOVL #8,W^CURRENT_TC
                                DD 030E          PUSHL #0
                                FB 0310          CALLS #1,W^REG_SAVE
                                0315 594          $CREPRC_S PIDADR = W^PID1,-
                                0315 595          IMAGE = W^IMAGE_NAME,-
                                0315 596          INPUT = W^IN,-
                                0315 597          OUTPUT = W^OUT,-
                                0315 598          ERROR = W^ERR,-
                                0315 599          BASPRI = #1,-
                                0315 600          PRIVADR = W^PRIVS,-
                                0315 601          MBXUNT = W^MBXUN,-
                                0315 602          QUOTA = W^QUOTA_LIST ; try all that
                                0347 603          FAIL_CHECK SSS_NORMAL ; check success
                                DD 0347          PUSHL #SS$ NORMAL
                                FB 034D          CALLS #1,W^REG_CHECK
021F'CF 0175'CF  DE 0352 604          MOVAL W^GETJPI,W^SERV_NAME ; set service name
                                0359 605          $GETJPI_S PIDADR = W^PID1,-
                                0359 606          ITMLST = W^GET_LIST ; get the base priority
                                0370 607          FAIL_CHECK SSS_NORMAL ; check success
                                DD 0370          PUSHL #SS$ NORMAL
                                FB 0376          CALLS #1,W^REG_CHECK
021F'CF 016E'CF  DE 037B 608          MOVAL W^CREPRC,W^SERV_NAME ; set service name
                                01C3'CF 01  D1 0382 609          CMPL #1,W^PRIB ; is it correct?
                                13 0387 610          BEQL 20$ ; br if OK
                                DD 0389 611          PUSHAL W^PRIB ; push received
                                DD 038D 612          PUSHAL #1 ; push expected
```



```
0004'CF 0A DO 049C
00 DD 049C
0BF3'CF 01 FB 04A1
04A3 661
04A8 662
04A8 663
04A8 664
04A8 665
04A8 666
04A8 667
04A8 668
04A8 669
04A8 670
04DC 671
00000000'8F DD 04DC
0BFD'CF 01 FB 04E2
021F'CF 0175'CF DE 04E7 672
04EE 673
04EE 674
0505 675
00000000'8F DD 0505
0BFD'CF 01 FB 050B
021F'CF 016E'CF DE 0510 676
0565'CF 003C'CF D1 0517 677
11 13 051E 678
003C'CF DD 0520 679
0565'CF DD 0524 680
0144'CF DF 0528 681
0C3F'CF 03 FB 052C 682
0531 683
0531 684
0D30'CF 00 FB 053E 685
0543 686
0543 687
0543 688
0543 689
0543 690
0543 691
0004'CF 0B DO 0543
00 DD 0548
0BF3'CF 01 FB 054A
054F 692
59 00000000'9F DO 056C 693
0051'CF 69 DE 0573 694
0578 695
0579 696
0599 697
05B9 698
00 DD 05D9 699
0BFD'CF 01 FB 05DB 700
021F'CF 0175'CF DE 05E0 701
05E7 702
05FC 703

STP10:
MOVL #10,W^CURRENT_TC
PUSHL #0
CALLS #1,W^REG_SAVE
$CREPRC_S PIDADR = W^PIDT,-
IMAGE = W^IMAGE_NAME,-
INPUT = W^IN,-
OUTPUT = W^OUT,-
ERROR = W^ERR,-
BASPRI = #2,-
PRVADR = W^PRIVS,-
MBXUNT = W^MBXUN,-
QUOTA = W^QUOTA_LIST,-
UIC = W^PROC_UIC
FAIL_CHECK SSS_NORMAL
PUSHL #SS$ NORMAL
CALLS #1,W^REG_CHECK
MOVAL W^GETJPI,W^SERV_NAME
$GETJPI_S PIDADR = W^PID1,-
ITMLST = W^GET_LIST
FAIL_CHECK SSS_NORMAL
PUSHL #SS$ NORMAL
CALLS #1,W^REG_CHECK
MOVAL W^CREPRC,W^SERV_NAME
CMPL W^UIC,W^PROC_UIC
BEQL 40$
PUSHL W^UIC
PUSHL W^PROC_UIC
PUSHAL W^UIC MSG
CALLS #3,W^PRINT_FAIL
; try S and all this
; check success

; set service name
; get the process UIC
; check success

; reset the service name
; is the UIC correct?
; br if OK
; push received
; push expected
; push the string variable
; print the failure

; cause process termination
; check the process exit status

40$:
;+
; test the STSFLG's _S with all set
;-
NEXT_TEST

STP11:
MOVL #11,W^CURRENT_TC
PUSHL #0
CALLS #1,W^REG_SAVE
MODE TO,45$,KRNL,NOREGS
MOVL @#CTL$GL PHD,R9
MOVAL PHD$Q PRIVMSK(R9),W^PRIVMASK
MODE FROM,45$
PRIV ADD,PSWAPM
PRIV ADD,NOACNT
PRIV ADD,NETMBX
PUSHL #0
CALLS #1,W^REG_SAVE
MOVAL W^GETJPI,W^SERV_NAME
$GETJPI_S ITMLST = W^GET_LIST
FAIL_CHECK SSS_NORMAL
; kernal mode to access PHD
; get process header address
; get priv mask address
; get back to user mode
; add PSWAPM priv
; add NOACNT priv
; add NETMBX priv
; push a dummy param
; save a reg snap shot
; set service name
; get the current process privs
; check success
```



```
00000000'8F DD 05FC
0BFD'CF 01 FB 0602
021F'CF 016E'CF DE 0607 704
02B3'CF 0010'CF 7D 060E 705
0615 706
0615 707
0615 708
0615 709
0615 710
0615 711
0615 712
0615 713
0615 714
0615 715
0615 716
064D 717
00000000'8F DD 064D
0BFD'CF 01 FB 0653
021F'CF 0175'CF DE 0658 718
065F 719
065F 720
0676 721
00000000'8F DD 0676
0BFD'CF 01 FB 067C
021F'CF 016E'CF DE 0681 722
000001CF'EF FFC739FF 8F CA 0688 723
0693 724
0038C600 8F 01CF'CF D1 0693 725
13 13 069C 726
01CF'CF DD 069E 727
0038C600 8F DD 06A2 728
0134'CF DF 06A8 729
0C3F'CF 03 FB 06AC 730
06B1 731 50$:
06B1 732
06BE 733
0D30'CF 00 FB 06C0 734
56 00000000'8F D0 06C5 735
06CC 736 :+
06CC 737 :
06CC 738 : test the STSFLG's _G all clear
06CC 739 :
06CC 740 :-
06CC 741
NEXT_TEST
STP12:
0004'CF 0C D0 06CC
00 00 DD 06D1
0BF3'CF 01 FB 06D3
00A3'CF 02B3'CF DE 06D8 742
00AB'CF 054E'CF DE 06DF 743
00AF'CF 02 D0 06E6 744
00B3'CF 0565'CF D0 06EB 745
06F2 746
021F'CF 0175'CF DE 06FB 747
0702 748
0702 749

PUSHL #SS$ NORMAL
CALLS #1,W^REG CHECK
MOVAL W^CREPRC,W^SERV_NAME ; set the service name
MOVQ W^CURPRIV,W^PRIVS ; set the detached process privs
$CREPRC_S PIDADR = W^PID1,-
IMAGE = W^IMAGE_NAME,-
INPUT = W^IN,-
OUTPUT = W^OUT,-
ERROR = W^ERR,-
BASPRI = #2,-
PRVADR = W^PRIVS,-
MBXUNT = W^MBXUN,-
QUOTA = W^QUOTA_LIST,-
UIC = W^PROC_OIC,-
STSFLG = #XFF ; try every thing _S
; check success
FAIL_CHECK SS$ NORMAL
PUSHL #SS$ NORMAL
CALLS #1,W^REG CHECK
MOVAL W^GETJPI,W^SERV_NAME ; set service name
$GETJPI_S PIDADR = W^PID1,-
ITMLST = W^GET_LIST ; get process status flags
; check success
FAIL_CHECK SS$ NORMAL
PUSHL #SS$ NORMAL
CALLS #1,W^REG CHECK
MOVAL W^CREPRC,W^SERV_NAME ; reset service name
BICL #JPI_STS_NMASK,STS ; clear out any extraneous
; bits set by the Swapper
; flags OK?
; br if OK
; push received
; push expected
; push str variable
; print the failure
CMPL W^STS,#JPI_STS_MASK
BEQL 50$
PUSHL W^STS
PUSHL #JPI_STS_MASK
PUSHAL W^STSFLGS
CALLS #3,W^PRINT_FAIL
$DELPRI_S PIDADR = W^PID1
CLRL R6
CALLS #0,W^CRE CHECK
MOVL #SS$ NORMAL,R6 ; needed for bit 5 in STS
; set expected status return
; check image exit status
; reset the expected status return
;+
; test the STSFLG's _G all clear
;-
NEXT_TEST
STP12:
MOVL #12,W^CURRENT_TC
PUSHL #0
CALLS #1,W^REG SAVE
MOVAL W^PRIVS,W^CRE+CREPRC$ PRVADR ; setup PRVADR parameter
MOVAL W^PROC NAME,W^CRE+CREPRC$ PRCNAM ; setup PRCNAM parameter
MOVL #2,W^CRE+CREPRC$ BASPRI ; setup BASPRI parameter
MOVL W^PROC UIC,W^CRE+CREPRC$ UIC ; setup UIC parameter
$CREPRC_G W^CRE ; try it all _G
MOVAL W^GETJPI,W^SERV_NAME ; set service name
$GETJPI_S PIDADR = W^PID1,-
ITMLST = W^GET_LIST ; get the process status flags
```

```
00000000'8F DD 0719 750 FAIL_CHECK SSS_NORMAL ; check success
0BFD'CF 01 FB 071F PUSHL #SSS_NORMAL
021F'CF 016E'CF DE 0724 751 CALLS #1,W^REG_CHECK
000001CF'EF FFC739FF 8F CA 072B 752 MOVAL W^CREPRC,W^SERV_NAME ; set the service name
                                0736 753 BICL #JPI_STS_NMASK,STS ; clear out any extraneous
                                0736 754 CMPL W^STS,#^X0 ; bits set by the Swapper
                                073B 755 BEQL 60$ ; are they all OK?
                                073D 756 PUSHL W^STS ; br if OK
                                0741 757 PUSHL #^X0 ; push received
                                0743 758 PUSHAL W^STSFLGS ; push expected
0C3F'CF 03 FB 0747 759 CALLS #3,W^PRINT_FAIL ; push str variable
                                074C 760 60$: ; print the failure
                                074C 761 $WAKE_S PIDADR = W^PID1 ; cause process termination
0D30'CF 00 FB 0759 762 CALLS #0,W^CRE_CHECK ; check image exit status
```



```
075E 764 .SBTTL GETJPI TESTS
075E 765 :+
075E 766 :
075E 767 : $GETJPI tests
075E 768 :
075E 769 : test the default case with all items _S
075E 770 :
075E 771 :-
075E 772 : NEXT_TEST
075E
075E STP13:
075E
075E      MOVL    #13,W^CURRENT_TC
0763      PUSHL   #0
0765      CALLS   #1,W^REG_SAVE
076A      MOVAL   W^GETJPI,W^SERV_NAME      ; set service name
0771      MOVAL   W^UM,W^MODE                ; set the mode
0778      PRIV    ADD,SETPRV                 ; get ready to set privileges
0798      $SETPRV_S ENBFLG = #1,-           ; to known set for testing
0798      PRVADR = PRVMASK                  ; enable expected privileges
07AB      $SETPRV_S PRVADR = NPRVMASK
07BE      780
07BE      781      FAIL_CHECK SSS_NORMAL
07BE      PUSHL   #SS$ NORMAL
07C4      CALLS   #1,W^REG_CHECK
07C9      782      $GETJPI S ITMLST = W^GET_LIST      ; try S
07DE      783      FAIL_CHECK SSS_NORMAL      ; check success
07DE      PUSHL   #SS$ NORMAL
07E4      CALLS   #1,W^REG_CHECK
07E9      784      MOVAL   W^ACCOUNT,R6          ; set questionable data adr
07EE      785      MOVAL   W^JPI_GOOD,R7        ; set good data adr
58      00000044 8F DO 07F3 786      MOVL    #JPI_CIST_SIZE,R8      ; set the byte count
ODDA'CF 00 FB 07FA 787      CALLS   #0,W^JPI_CHECK      ; check the results
07FF 788 :+
07FF 789 :
07FF 790 : test _G default case with all items
07FF 791 :
07FF 792 :-
07FF 793 : NEXT_TEST
07FF
07FF STP14:
07FF
07FF      MOVL    #14,W^CURRENT_TC
0804      PUSHL   #0
0806      CALLS   #1,W^REG_SAVE
080B      794      $GETJPI G W^GET1
0814      795      FAIL_CHECK SSS_NORMAL
0814      PUSHL   #SS$ NORMAL
081A      CALLS   #1,W^REG_CHECK
081F      796      CALLS   #0,W^JPI_CHECK      ; check the results
0824 797 :+
0824 798 :
0824 799 : test local EFN
0824 800 :
0824 801 :-
0824 802 : NEXT_TEST
0824
0824 STP15:
```

```
0004'CF 0F DO 0824          MOVL #15,W^CURRENT_TC
                                PUSHL #0
0BF3'CF 01 FB 0829          CALLS #1,W^REG_SAVE
021F'CF 016E'CF DE 082B      MOVAL W^CREPRC,W^SERV_NAME ; set service name
                                $CREPRC_S QUOTA = W^QUOTA_LIST,-
                                IMAGE = W^IMAGE_NAME,-
                                PIDADR = W^PID1,-
                                PRCNAM = W^PROC_NAME
                                FAIL_CHECK SSS_NORMAL ; create the target process
                                PUSHL #SS$ NORMAL ; check for success
                                CALLS #1,W^REG_CHECK
00000000'8F DD 0861          MOVAL W^GETJPI,W^SERV_NAME ; reset the service name
0BFD'CF 01 FB 0867          $GETJPI_S EFN = #1,-
021F'CF 0175'CF DE 086C      ITMLST = W^GET_LIST ; try S with EFN
                                FAIL_CHECK SSS_NORMAL ; check success
                                PUSHL #SS$ NORMAL
                                CALLS #1,W^REG_CHECK
00000000'8F DD 0888          $WAITFR_S EFN = #1 ; wait for completion
0BFD'CF 01 FB 088E          CALLS #0,W^JPI_CHECK ; check the results
0DDA'CF 00 FB 089C          MOVL #1,W^GET+GETJPI$ EFN ; set the EFN
00C7'CF 01 DO 08A1          MOVAL W^PID1,W^GET+GETJPI$_PIDADR ; set the target process PID
00CB'CF 023B'CF DE 08A6      $GETJPI G W^GET ; try G with target process
                                FAIL_CHECK SSS_NORMAL ; check success
                                PUSHL #SS$ NORMAL
                                CALLS #1,W^REG_CHECK
00000000'8F DD 08B6          $WAITFR_S EFN = #1 ; wait for completion
0BFD'CF 01 FB 08BC          CALLS #0,W^JPI_CHECK ; check the results
0DDA'CF 00 FB 08CA          ;+
                                821 :+
                                822 : test common EFN with _S
                                823 :-
                                824 :-
                                825 :-
                                826
                                NEXT_TEST
                                STP16:
0004'CF 10 DO 08CF          MOVL #16,W^CURRENT_TC
0BF3'CF 01 FB 08D4          PUSHL #0
                                CALLS #1,W^REG_SAVE
                                $ASCEFC_S EFN = #65,-
                                NAME = W^EFC_NAME ; get a common EF
                                $GETJPI_S EFN = #65,-
                                ITMLST = W^GET_LIST ; try S with CEFN
                                FAIL_CHECK SSS_NORMAL ; check success
                                PUSHL #SS$ NORMAL
                                CALLS #1,W^REG_CHECK
00000000'8F DD 0909          $WAITFR_S EFN = #65 ; wait for completion
0BFD'CF 01 FB 090F          CALLS #0,W^JPI_CHECK ; check results
0DDA'CF 00 FB 0914          MOVL #65,W^GET+GETJPI$_EFN ; set the common EFC
00C7'CF 00000041 8F DO 0926      $GETJPI G W^GET ; try G, CEFN, and target process
                                FAIL_CHECK SSS_NORMAL ; check for success
                                PUSHL #SS$ NORMAL
                                CALLS #1,W^REG_CHECK
                                $WAITFR_S EFN = #65 ; wait for completion
                                CALLS #0,W^JPI_CHECK ; check the results
                                $DACEFC_S EFN = #65 ; release the CEFN
                                832
                                833
                                834
                                835
                                836
                                837
                                838
                                839
                                840
                                841 ;+
```



```
0962 842 : test PIDADR
0962 843 :
0962 844 :
0962 845 :-
0962 846 NEXT_TEST
0962
0962 STP17:
0004'CF 11 DO 0962 MOVL #17,W^CURRENT_TC
00 00 DD 0967 PUSHL #0
0BF3'CF 01 FB 0969 CALLS #1,W^REG_SAVE
096E 847 $GETJPI_S EFN = #2,-
096E 848 PIDADR = W^PID1,-
096E 849 ITMLST = W^GET_LIST ; try S with PID
0985 850 FAIL_CHECK SSS_NORMAL ; check success
00000000'8F DD 0985 PUSHL #SS$ NORMAL
0BFD'CF 01 FB 098B CALLS #1,W^REG_CHECK
0990 851 $WAITFR_S EFN = #2 ; wait for completion
0DDA'CF 00 FB 0999 852 CALLS #0,W^JPI_CHECK ; check the results
099E 853 :+
099E 854 :
099E 855 : test PRCNAM
099E 856 :
099E 857 :-
099E 858 NEXT_TEST
099E
099E STP18:
0004'CF 12 DO 099E MOVL #18,W^CURRENT_TC
00 00 DD 09A3 PUSHL #0
0BF3'CF 01 FB 09A5 CALLS #1,W^REG_SAVE
09AA 859 $GETJPI_S EFN = #3,-
09AA 860 PRCNAM = W^PROC_NAME,-
09AA 861 ITMLST = W^GET_LIST ; try S with PRCNAM
09C1 862 FAIL_CHECK SSS_NORMAL ; check success
00000000'8F DD 09C1 PUSHL #SS$ NORMAL
0BFD'CF 01 FB 09C7 CALLS #1,W^REG_CHECK
09CC 863 $WAITFR_S EFN = #3 ; wait for completion
0DDA'CF 00 FB 09D5 864 CALLS #0,W^JPI_CHECK ; check the results
09DA 865 $GETJPI_S EFN = #16,-
09DA 866 PRCNAM = W^TEST_MOD_NAME_D,-
09DA 867 ITMLST = W^GET_LIST ; try S with PRCNAM on self
09F1 868 FAIL_CHECK SSS_NORMAL ; check success
00000000'8F DD 09F1 PUSHL #SS$ NORMAL
0BFD'CF 01 FB 09F7 CALLS #1,W^REG_CHECK
09FC 869 $WAITFR_S EFN = #16 ; wait for completion
0DDA'CF 00 FB 0A05 870 CALLS #0,W^JPI_CHECK ; check the results
0A0A 871 :+
0A0A 872 :
0A0A 873 : test IOSB
0A0A 874 :
0A0A 875 :-
0A0A 876 NEXT_TEST
0A0A
0A0A STP19:
0004'CF 13 DO 0A0A MOVL #19,W^CURRENT_TC
00 00 DD 0A0F PUSHL #0
0BF3'CF 01 FB 0A11 CALLS #1,W^REG_SAVE
0A16 877 $GETJPI_S EFN = #4,-
```

```
00000000'8F DD 0A16 878 IOSB = W^IOSTAT,-
OBFD'CF 01 FB 0A16 879 PRCNAM = W^IMAGE_NAME,-
00000000'8F DD 0A16 880 PIDADR = W^PID1,-
OBFD'CF 01 FB 0A16 881 ITMLST = W^GET_LIST ; try all this stuff
ODDA'CF 00 FB 0A31 882 FAIL_CHECK SS$ NORMAL ; check success
00000000'8F DD 0A31 883 PUSHL #SS$ NORMAL
OBFD'CF 01 FB 0A37 884 CALLS #1,W^REG_CHECK
ODDA'CF 00 FB 0A3C 885 $WAITFR_S EFN = #4 ; wait for completion
00000000'8F DD 0A45 886 CALLS #0,W^JPI_CHECK ; check the results
OBFD'CF 01 FB 0A4A 887 :+
00000000'8F DD 0A4A 888 : test ASTADR and ASTPRM _S
OBFD'CF 01 FB 0A4A 889 :-
00000000'8F DD 0A4A 890 NEXT_TEST
OBFD'CF 01 FB 0A4A 891 STP20:
00000000'8F DD 0A4A 892 MOVL #20,W^CURRENT_TC
OBFD'CF 01 FB 0A4F 893 PUSHL #0
00000000'8F DD 0A51 894 CALLS #1,W^REG_SAVE
OBFD'CF 01 FB 0A56 895 $SETAST_S ENBFLG = #0 ; disable AST's
00000000'8F DD 0A5F 896 $GETJPI_S ASTADR = B^20$,-
OBFD'CF 01 FB 0A5F 897 ASTPRM = #5,-
00000000'8F DD 0A5F 898 IOSB = W^IOSTAT,-
OBFD'CF 01 FB 0A5F 899 PRCNAM = W^IMAGE_NAME,-
00000000'8F DD 0A5F 900 PIDADR = W^PID1,-
OBFD'CF 01 FB 0A5F 901 ITMLST = W^GET_LIST ; try an AST
00000000'8F DD 0A7D 898 FAIL_CHECK SS$ NORMAL ; check success
OBFD'CF 01 FB 0A7D 899 PUSHL #SS$ NORMAL
00000000'8F DD 0A83 900 CALLS #1,W^REG_CHECK
OBFD'CF 01 FB 0A88 899 $SETAST_S ENBFLG = #1 ; let er rip
00000000'8F DD 0A91 900 $HIBER_S ; wait here for completion
00000000'8F DD 0A98 901 BRB 40$ ; jump over the AST routine
00000000'8F DD 0A9A 902 20$:
00000000'8F DD 0A9A 903 .WORD ^M<R2,R3,R4>
00000000'8F DD 0A9C 904 CMPL #5,4(AP) ; is this the right param
00000000'8F DD 0AA0 905 BEQL 30$ ; br if OK
00000000'8F DD 0AA2 906 PUSHL 4(AP) ; push the received
00000000'8F DD 0AA5 907 PUSHL #5 ; push expected
00000000'8F DD 0AA7 908 PUSHAL W^AST_PARAM ; push the string variable
00000000'8F DD 0AAB 909 CALLS #3,W^PRINT_FAIL ; print the failure
00000000'8F DD 0AB0 910 30$:
00000000'8F DD 0AB0 911 CALLS #0,W^JPI_CHECK ; check the results
00000000'8F DD 0AB5 912 $WAKE_S ; time to wake up
00000000'8F DD 0AC0 913 RET ; return
00000000'8F DD 0AC1 914 40$:
00000000'8F DD 0AC1 915 :+
00000000'8F DD 0AC1 916 : test ASTADR and ASTPRM _G to test all offset definitions
00000000'8F DD 0AC1 917 :-
00000000'8F DD 0AC1 918 :
00000000'8F DD 0AC1 919 :
00000000'8F DD 0AC1 920 NEXT_TEST
00000000'8F DD 0AC1 921 STP21:
00000000'8F DD 0AC6 922 MOVL #21,W^CURRENT_TC
00000000'8F DD 0AC8 923 PUSHL #0
00000000'8F DD 0AC8 924 CALLS #1,W^REG_SAVE
```


Address	Hex	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op46
---------	-----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

SATSSS35
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00 Page 26
GETJPI TESTS 5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1 (1)

				OBCC	967	TEST_END	
	004C'CF	DD		OBCC		PUSHL	W^TMD_ADDR
	0048'CF	DD		OBDO		PUSHL	W^TMN_ADDR
	02	DD		OBDA		PUSHL	#2
	0044'CF	DD		OBDA		PUSHL	W^MOD_MSG_CODE
00000000'GF	04	FB		OBDA		CALLS	\$\$T1,G^LIBSSIGNAL
0044'CF	01	1C	01	FO	OBE1	INSV	#1,\$STSV_INHIB_MSG,#1,W^MOD_MSG_CODE
	0044'CF	DD		OBE8		PUSHL	W^MOD_MSG_CODE
00000000'GF	01	FB		OBE8		CALLS	#1,G^SYS\$EXIT


```

OBF3 969 .SBTTL ROUTINES
OBF3 970 .SBTTL REG_SAVE
OBF3 971 :++
OBF3 972 : FUNCTIONAL DESCRIPTION:
OBF3 973 : Subroutine to save R2-R11 in the register save location.
OBF3 974 :
OBF3 975 : CALLING SEQUENCE:
OBF3 976 :     PUSHL #0 ; save a dummy parameter
OBF3 977 :     CALLS #1,W^REG_SAVE ; save R2-R11
OBF3 978 :
OBF3 979 : INPUT PARAMETERS:
OBF3 980 :     NONE
OBF3 981 :
OBF3 982 : OUTPUT PARAMETERS:
OBF3 983 :     NONE
OBF3 984 :
OBF3 985 :--
OBF3 986
OBF3 987 REG_SAVE:
OBF3 988 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
OBF5 989 MOVC3 #4*10,^X14(FP),W^REG_SAVE_AREA ; save the registers in the program
OBF3 990 RET
OBF3 991 .SBTTL REG_CHECK
OBF3 992 :++
OBF3 993 : FUNCTIONAL DESCRIPTION:
OBF3 994 : Subroutine to test R0 & R2-R11 for proper content after a service
OBF3 995 : execution. A snapshot is taken by the REG_SAVE routine at the
OBF3 996 : beginning of each step and this routine is executed after the
OBF3 997 : services have been executed.
OBF3 998 :
OBF3 999 : CALLING SEQUENCE:
OBF3 1000 :     PUSHL #SS$ XXXXXX ; push expected R0 contents
OBF3 1001 :     CALLS #1,W^REG_CHECK ; execute this routine
OBF3 1002 :
OBF3 1003 : INPUT PARAMETERS:
OBF3 1004 :     expected R0 contents on the stack
OBF3 1005 :
OBF3 1006 : OUTPUT PARAMETERS:
OBF3 1007 :     possible error messages printed using $PUTMSG
OBF3 1008 :
OBF3 1009 :--
OBF3 1010
OBF3 1011 REG_CHECK:
OBF3 1012 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
OBF3 1013 CMPL 4(AP),R0 ; is this the right fail code?
OBF3 1014 BEQL 10$ ; br if yes
OBF3 1015 PUSHL R0 ; push received data
OBF3 1016 PUSHL 4(AP) ; push expected data
OBF3 1017 PUSHAL W^EXP ; push the string variable
OBF3 1018 CALLS #3,W^PRINT_FAIL ; print the error message
OBF3 1019 10$:
OBF3 1020 CMPC3 #4*10,^X14(FP),W^REG_SAVE_AREA ; check all but R0
OBF3 1021 BEQL 20$ ; br if O.K.
OBF3 1022 SUBL3 #REG_SAVE_AREA,R3,R6 ; calculate the register number
OBF3 1023 DIVL2 #4,R6
OBF3 1024 ADDB3 #^X2,R6,-(SP) ; set number past R0-R1 and save
OBF3 1025 BICL2 #3,R1 ; backup to register boundrys

0008'CF 14 AD 28 OFFC 28 04
50 04 AC D1 OBF3 1012
OE 13 OBF3 1013
50 DD OBF3 1014
04 AC DD OBF3 1015
00D8'CF DF OBF3 1016
OC3F'CF 03 FB OBF3 1017
OBF3 1018
OBF3 1019
0008'CF 14 AD 28 29 OBF3 1020
56 53 00000008'8F C3 OBF3 1021
56 04 C6 OBF3 1022
7E 56 02 81 OBF3 1023
51 03 CA OBF3 1024
OBF3 1025
```



```
53 03 CA 0C2E 1026 BICL2 #3,R3
61 DD 0C31 1027 PUSHL (R1) ; push received data
63 DD 0C33 1028 PUSHL (R3) ; push expected data
006D'CF DF 0C35 1029 PUSHAL W^REG ; set string pntr param.
0C3F'CF 04 FB 0C39 1030 CALLS #4,W^PRINT_FAIL ; print the error message
04 0C3E 1031 20$: RET
0C3E 1032 .SBTTL PRINT_FAIL
0C3F 1033 :++
0C3F 1034 : FUNCTIONAL DESCRIPTION:
0C3F 1035 : Subroutine to report failures using $PUTMSG
0C3F 1036 :
0C3F 1037 : CALLING SEQUENCE:
0C3F 1038 : Mode #1 PUSHL EXPECTED Mode #2 PUSHL REG NUMBER
0C3F 1039 : PUSHL RECEIVED PUSHL EXPECTED
0C3F 1040 : PUSHAL STRING VAR PUSHL RECEIVED
0C3F 1041 : CALLS #3,W^PRINT_FAIL PUSHAL STRING VAR
0C3F 1042 : CALLS #4,W^PRINT_FAIL
0C3F 1043 :
0C3F 1044 : Mode #3 PUSHAL STRING VAR
0C3F 1045 : CALLS #1,W^PRINT_FAIL
0C3F 1046 :
0C3F 1047 : INPUT PARAMETERS:
0C3F 1048 : listed above
0C3F 1049 :
0C3F 1050 : OUTPUT PARAMETERS:
0C3F 1051 : an error message is printed using $PUTMSG
0C3F 1052 :
0C3F 1053 :--
0C3F 1054 :
003C 0C3F 1055 PRINT_FAIL:
0C41 1056 .WORD ^M<R2,R3,R4,R5>
0C62 1057 $FAO_S W^CS1,W^MESSAGEL,W^MSGL,#TEST_MOD_NAME,W^SERV_NAME,W^CURRENT_TC
0C73 1058 $PUTMSG_S W^MSGVEC ; print the message
0C76 1059 CMPB (AP),#4 ; is this a register message?
0C78 1060 BEQL 10$ ; br if yes
0C7B 1061 CMPB (AP),#1 ; is this just a message?
0C7D 1062 BEQL 20$ ; br if yes
0C9C 1063 $FAO_S W^CS2,W^MESSAGEL,W^MSGL,4(AP),8(AP),4(AP),12(AP)
0C9E 1064 BRB 30$ ; goto output message
0C9E 1065 10$: $FAO_S W^CS3,W^MESSAGEL,W^MSGL,4(AP),16(AP),8(AP),4(AP),16(AP),12(AP)
0CC3 1066 BRB 30$ ; goto output message
0CC5 1067 20$:
0CC5 1068 MOVL 4(AP),W^MSGVEC1+12 ; save string address
0CCB 1069 $PUTMSG_S W^MSGVEC1 ; print the message
0CDC 1070 BRB 40$ ; skip the other message
0CDE 1071 30$:
0CDE 1072 $PUTMSG_S W^MSGVEC ; print the message
0CEF 1073 40$:
0CEF 1074 CALLS #0,W^MODE_ID ; identify the mode
0CF4 1075 MOVAL W^TEST_MOD_FAIL,W^TMD_ADDR ; set failure message address
0CFB 1076 INSV #ERROR,#0,#3,W^MOD_MSG_CODE ; set severity code
04 0D02 1077 RET
```



```
0D03 1080 .SBTTL MODE_ID
0D03 1081 :++
0D03 1082 : FUNCTIONAL DESCRIPTION:
0D03 1083 : Subroutine to identify the mode that an exit handler is in.
0D03 1084 :
0D03 1085 : CALLING SEQUENCE:
0D03 1086 : CALLS #0,W^MODE_ID
0D03 1087 :
0D03 1088 : INPUT PARAMETERS:
0D03 1089 : MODE contains an address pointing to an ascii string desc.
0D03 1090 : of the current CPU mode.
0D03 1091 :
0D03 1092 : OUTPUT PARAMETERS:
0D03 1093 : NONE
0D03 1094 :
0D03 1095 :--
0D03 1096 :
003C 0D03 1097 MODE_ID:
0D03 1098 .WORD ^M<R2,R3,R4,R5>
0D05 1099 $FAO S W^CS5,W^MESSAGEL,W^MSGL,MODE ; format the error message
0D1E 1100 $PUTMSG_S W^MSGVEC ; print the mode message
04 0D2F 1101 RET
0D30 1102 .SBTTL CRE_CHECK
0D30 1103 :++
0D30 1104 : FUNCTIONAL DESCRIPTION:
0D30 1105 : Routine to check the process exit status of a created process.
0D30 1106 :
0D30 1107 : CALLING SEQUENCE:
0D30 1108 : CALLS #0,W^CRE_CHECK ; save R2-R11
0D30 1109 :
0D30 1110 : INPUT PARAMETERS:
0D30 1111 : R6 = Expected process exit status
0D30 1112 : R7 = PID check flag BIT0 = 1 means check the PID
0D30 1113 :
0D30 1114 : OUTPUT PARAMETERS:
0D30 1115 : NONE
0D30 1116 :
0D30 1117 :--
0D30 1118 :
OFFC 0D30 1119 CRE_CHECK:
0D30 1120 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0D32 1121 $QIOW_S EFN = #1,-
0D32 1122 FUNC = #10$, READVBLK,-
0D32 1123 CHAN = W^MBCHAN,-
0D32 1124 IOSB = W^IOSTATUS,-
0D32 1125 P1 = W^MBUF,-
0D32 1126 P2 = #100
16 0D59 1127 CMPL W^MBUF+ACCSL_FINALSTS,R6 ; read the mail
0F 0D5E 1128 BEQL 10$ ; is the status as expected?
024F'CF 0F 13 0D60 1129 PUSHL W^MBUF+ACCSL_FINALSTS ; br if OK
56 0D64 1130 PUSHL R6 ; push received
00D8'CF DF 0D66 1131 PUSHAL W^EXP ; push expected
FEDO CF 03 FB 0D6A 1132 CALLS #3,W^PRINT_FAIL ; push string variable
; print the failure
1A 57 E9 0D6F 1133 10$:
0247'CF 023B'CF D1 0D72 1134 BLBC R7,20$ ; should we check the PID?
11 13 0D79 1135 CMPL W^PID1,W^IOSTATUS+4 ; check the PID
; br if its good
BEQL 20$
```

```
0247'CF DD 0D7B 1137          PUSHL  W^IOSTATUS+4          ; push received
023B'CF DD 0D7F 1138          PUSHL  W^PID1              ; push expected
0163'CF DF 0D83 1139          PUSHAL W^PID STR           ; push the string variable
FEB3 CF 03 FB 0D87 1140          CALLS  #3,W^PRINT_FAIL    ; print the failure
                                20$:
                                04 0D8C 1141          RET
                                0D8C 1142          .SBTTL JPI_CHECK
                                0D8D 1143          :++
                                0D8D 1144          : FUNCTIONAL DESCRIPTION:
                                0D8D 1145          : Subroutine to check the results of a JPI service
                                0D8D 1146          :
                                0D8D 1147          : CALLING SEQUENCE:
                                0D8D 1148          : CALLS  #0,W^JPI_CHECK ; check the results
                                0D8D 1149          :
                                0D8D 1150          : INPUT PARAMETERS:
                                0D8D 1151          : R6 = questionable data address
                                0D8D 1152          : R7 = good data address
                                0D8D 1153          : R8 = byte count
                                0D8D 1154          :
                                0D8D 1155          : OUTPUT PARAMETERS:
                                0D8D 1156          : NONE
                                0D8D 1157          :
                                0D8D 1158          : --
                                0D8D 1159          :
                                0D8D 1160          :
                                0D8D 1161          : ARGLST1:
                                0D8D 1162          : .BLKL 3
                                0D99 1163          : CTRSTR:
                                0D99 1164          : .ASCID /data error at offset !XW, good data = !XB bad data = !XB./

65 20 61 74 61 64 00000DA1'010E0000' 0D99 1165          :
73 66 66 6F 20 74 61 20 72 6F 72 72 0DA7 1166          : JPI_CHECK:
64 6F 6F 67 20 2C 57 58 21 20 74 65 0DB3 1167          : .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
20 42 58 21 20 3D 20 61 74 61 64 20 0DBF 1168          : CMPC3 R8,(R6),(R7) ; check the buffer
21 20 3D 20 61 74 61 64 20 64 61 62 0DCB 1169          : BEQL 10$ ; br if good
                                2E 42 58 0DD7 1170          : SUBL3 #ACCOUNT,R1,W^ARGLST1 ; get buffer offset
                                0DDA 1165          :
                                0DDA 1166          : JPI_CHECK:
                                0DDA 1167          : .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                                0DDC 1168          : CMPC3 R8,(R6),(R7) ; check the buffer
                                0DE0 1169          : BEQL 10$ ; br if good
                                0DE2 1170          : SUBL3 #ACCOUNT,R1,W^ARGLST1 ; get buffer offset
                                0DEC 1171          : MOVZBL (R3),W^ARGLST1+4 ; get the good data
                                0DF1 1172          : MOVZBL (R1),W^ARGLST1+8 ; get the bad data
                                0DF6 1173          : $FAOL S W^CTRSTR,W^ML,W^GETBUF,W^ARGLST1 ; make it readable
                                0E0D 1174          : PUSHAL W^ML ; push the desc. address
                                0E11 1175          : CALLS #1,W^PRINT_FAIL ; print the failure
                                0E16 1176          :
                                04 0E16 1177          : RET
                                10$:
                                04 0E16 1177          : RET
```

00000D99

67 66 58 29 0FFC 0DDA 1167

FFA1 CF 51 00000000'8F C3 0DE0 1169

FFA0 CF 63 9A 0DEC 1171

FF9F CF 61 9A 0DF1 1172

0183'CF DF 0DF6 1173

FE29 CF 01 FB 0E0D 1174

04 0E11 1175

04 0E16 1176

04 0E16 1177


```
0E17 1180 MOD_MSG_PRINT:
0E17 1181 :
0E17 1182 : *****
0E17 1183 : *
0E17 1184 : * PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES *
0E17 1185 : * (USING THE PUTMSG MACRO). *
0E17 1186 : *
0E17 1187 : *****
0E17 1188 :
05 0E17 1189 PUTMSG <MOD_MSG_CODE,#2,TMN_ADDR,TMD_ADDR> : PRINT MSG
0E32 1190 RSB ; ... AND RETURN TO CALLER
0E33 1191 :
0E33 1192 CHMRTN:
0E33 1193 : *****
0E33 1194 : *
0E33 1195 : * CHANGE MODE ROUTINE. THIS ROUTINE GETS CONTROL WHENEVER *
0E33 1196 : * A CMKRNL, CMEXEC, OR CMSUP SYSTEM SERVICE IS ISSUED *
0E33 1197 : * BY THE MODE MACRO ('TO' OPTION). IT MERELY DOES *
0E33 1198 : * A JUMP INDIRECT ON A FIELD SET UP BY MODE. IT HAS *
0E33 1199 : * THE EFFECT OF RETURNING TO THE END OF THE MODE *
0E33 1200 : * MACRO EXPANSION. *
0E33 1201 : *
0E33 1202 : *****
0E33 1203 :
00000059'FF 0000 0E33 1204 .WORD 0 ; ENTRY MASK
17 0E35 1205 JMP @CHM_CONT ; RETURN TO MODE MACRO IN NEW MODE
0E3B 1206 :
0E3B 1207 : * RET INSTR WILL BE ISSUED IN EXPANSION OF 'MODE FROM, ....' MACRO
0E3B 1208 :
0E3B 1209 .END SATSSS35
```

SATSSS35
Symbol table

I 10
- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 32
(3)

\$\$ARGS	= 00000007		
\$\$T1	= 00000004		
\$\$T2	= 00000004		
ACCSL_FINALSTS	= 00000004		
ACCOUNT	00000000	R	02
ACCOUNTL	00000008	R	02
APTCNT	00000050	R	02
APTCNTL	00000054	R	02
ARGLST1	0000008D	R	05
ASTACT	00000056	R	02
ASTACTL	0000005A	R	02
ASTCNT	00000062	R	02
ASTCNTL	00000066	R	02
ASTEN	0000005C	R	02
ASTENL	00000060	R	02
ASTLM	00000068	R	02
ASTLML	0000006C	R	02
AST_PARAM	000000E6	R	03
AUTHPRIV	0000006E	R	02
AUTHPRIVL	00000076	R	02
BIOCNT	00000078	R	02
BIOCNTL	0000007C	R	02
BIOLM	0000007E	R	02
BIOLML	00000082	R	02
BP	000000F8	R	03
BUF	00000133	R	04
BUFIO	00000084	R	02
BUFIOI	00000088	R	02
BYTCNT	0000008A	R	02
BYTCNTL	0000008E	R	02
BYTLM	00000090	R	02
BYTLM	00000094	R	02
CHMRTN	00000E33	R	05
CHM_CONT	00000059	R	04
CPULIM	0000000A	R	02
CPULIML	0000000E	R	02
CPUTIM	00000096	R	02
CPUTIML	0000009A	R	02
CRE	0000008B	R	04
CREPRC	0000016E	R	03
CREPRCS_BASPRI	= 00000024		
CREPRCS_ERROR	= 00000014		
CREPRCS_IMAGE	= 00000008		
CREPRCS_INPUT	= 0000000C		
CREPRCS_ITMLST	= 00000034		
CREPRCS_MBXUNT	= 0000002C		
CREPRCS_NARGS	= 0000000D		
CREPRCS_OUTPUT	= 00000010		
CREPRCS_PIDADR	= 00000004		
CREPRCS_PRCNAM	= 00000020		
CREPRCS_PRVADR	= 00000018		
CREPRCS_QUOTA	= 0000001C		
CREPRCS_STSFLG	= 00000030		
CREPRCS_UIC	= 00000028		
CRE_CHECK	000000D30	R	05
CS1	00000031	R	03
CS2	00000063	R	03

CS3	00000090	R	03
CS5	000000C3	R	03
CTL\$GL_PHD	*****	X	05
CTRSTR	000000D99	R	05
CURPRIV	00000010	R	02
CURPRIVL	00000018	R	02
CURRENT_TC	00000004	R	04
DFPFC	0000009C	R	02
DFPFC	000000A0	R	02
DFWSCNT	000000A2	R	02
DFWSCNTL	000000A6	R	02
DIB\$W_UNIT	= 0000000C		
DIOCNT	000000A8	R	02
DIOCNTL	000000AC	R	02
DIOLM	000000AE	R	02
DIOLML	000000B2	R	02
DIRIO	000000B4	R	02
DIRIOL	000000B8	R	02
DIRTY	00000266	R	03
EFCS	000000BA	R	02
EFCSL	000000BE	R	02
EFCU	000000C0	R	02
EFCUL	000000C4	R	02
EFC_NAME	0000014F	R	03
EFWM	000000C6	R	02
EFWML	000000CA	R	02
ERR	00000529	R	03
ERROR	= 00000002		
EXCVEC	000000CC	R	02
EXCVECL	000000D0	R	02
EXP	000000D8	R	03
FILCNT	000000D8	R	02
FILCNTL	000000DC	R	02
FILLM	000000DE	R	02
FILLML	000000E2	R	02
FINALEXC	000000D2	R	02
FINALEXCL	000000D6	R	02
FREPOVA	000000E4	R	02
FREPOVAL	000000E8	R	02
FREP1VA	000000EA	R	02
FREP1VAL	000000EE	R	02
GET	000000C3	R	04
GET1	000000E3	R	04
GETBUF	0000018B	R	04
GETJPI	00000175	R	03
GETJPI\$ASTADR	= 00000018		
GETJPI\$ASTPRM	= 0000001C		
GETJPI\$EFN	= 00000004		
GETJPI\$IOSB	= 00000014		
GETJPI\$ITMLST	= 00000010		
GETJPI\$NARGS	= 00000007		
GETJPI\$PIDADR	= 00000008		
GETJPI\$PRCNAM	= 0000000C		
GET_LIST	000001EE	R	03
GPGCNT	000000F0	R	02
GPGCNTL	000000F4	R	02
GRP	0000001A	R	02

SATSSS35
Symbol table

J 10
- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 33
(3)

GRPL	0000001E	R	02	JPI\$-STATE	=	00000306		
IMAGE_NAME	0000053A	R	03	JPI\$-STS	=	00000305		
IMAGNAME	000000F6	R	02	JPI\$-TMBU	=	00000308		
IMAGNAMEL	00000176	R	02	JPI\$-TQCNT	=	00000315		
IMAGPRIV	00000020	R	02	JPI\$-TQLM	=	00000410		
IMAGPRIVL	00000028	R	02	JPI\$-UIC	=	00000304		
IN	00000506	R	03	JPI\$-USERNAME	=	00000202		
IOS_READVBLK	*****	X	05	JPI\$-VIRTPEAK	=	00000200		
IOSTAT	00000233	R	04	JPI\$-VOLUMES	=	00000205		
IOSTATUS	00000243	R	04	JPI\$-WSAUTH	=	00000401		
ITEM_LIST	00000103	R	04	JPI\$-WSPEAK	=	00000201		
JPI\$-ACCOUNT	=	00000203		JPI\$-WSQUOTA	=	00000402		
JPI\$-APTCNT	=	0000030A		JPI\$-WSSIZE	=	00000411		
JPI\$-ASTACT	=	00000300		JPI-CHECK	00000DDA	R	05	
JPI\$-ASTCNT	=	0000030E		JPI-GOOD	000004B6	R	03	
JPI\$-ASTEN	=	00000301		JPI-GOOD_SHRT	000004C0	R	03	
JPI\$-ASTLM	=	00000409		JPI-LIST-SIZE	=	00000044		
JPI\$-AUTHPRIV	=	00000412		JPI-LIST-SIZE1	=	0000003A		
JPI\$-BIOCNT	=	0000030F		JPI-PRV_MASK	=	1070BFEE		
JPI\$-BIOLM	=	00000310		JPI-PRV_NMASK	=	EF8F4010		
JPI\$-BUFIO	=	0000040C		JPI-STS_MASK	=	0038C600		
JPI\$-BYTCNT	=	00000311		JPI-STS_NMASK	=	FFC739FF		
JPI\$-BYTLM	=	0000031A		LIB\$SIGNAL	*****	X	05	
JPI\$-CPULIM	=	0000040D		LOGINTIM	00000178	R	02	
JPI\$-CPUTIM	=	00000407		LOGINTIML	0000017C	R	02	
JPI\$-CURPRIV	=	00000400		MBCHAN	0000023F	R	04	
JPI\$-DFPFC	=	00000406		MBNAM	00000188	R	03	
JPI\$-DFWSCNT	=	00000403		MBUF	00000248	R	04	
JPI\$-DIOCNT	=	00000312		MBXUN	00000241	R	04	
JPI\$-DIOLM	=	00000313		MEM	0000002A	R	02	
JPI\$-DIRIO	=	0000040B		MEML	0000002E	R	02	
JPI\$-EFCS	=	00000317		MESSAGEL	00000217	R	04	
JPI\$-EFCU	=	00000318		ML	00000183	R	04	
JPI\$-EFWM	=	00000316		MODE	00000069	R	04	
JPI\$-EXCVEC	=	00000100		MODE_ID	00000D03	R	05	
JPI\$-FILCNT	=	00000314		MOD_MSG_CODE	00000044	R	04	
JPI\$-FILLM	=	0000040F		MOD_MSG_PRINT	00000E17	R	05	
JPI\$-FINALEXC	=	00000101		MSGC	00000083	R	04	
JPI\$-FREPOVA	=	00000404		MSGVEC	000001A6	R	03	
JPI\$-FREPIVA	=	00000405		MSGVEC1	00000223	R	04	
JPI\$-GPCCNT	=	0000030C		NPRVMASK	0000019E	R	03	
JPI\$-GRP	=	00000308		OUT	00000517	R	03	
JPI\$-IMAGNAME	=	00000207		OWNER	0000017E	R	02	
JPI\$-IMAGPRIV	=	00000413		OWNERL	00000182	R	02	
JPI\$-LOGINTIM	=	00000206		PAGEFLTS	00000184	R	02	
JPI\$-MEM	=	00000307		PAGEFLTSL	00000188	R	02	
JPI\$-OWNER	=	00000303		PCB\$V_BATCH	=	0000000E		
JPI\$-PAGEFLTS	=	0000040A		PCB\$V_HIBER	=	00000013		
JPI\$-PGFLQUOTA	=	0000040E		PCB\$V_LOGIN	=	00000014		
JPI\$-PID	=	00000319		PCB\$V_NETWORK	=	00000015		
JPI\$-PPGCNT	=	0000030D		PCB\$V_NOACNT	=	0000000F		
JPI\$-PRCCNT	=	0000031B		PCB\$V_SSFEXCU	=	00000009		
JPI\$-PRCLM	=	00000408		PCB\$V_SSRWAIT	=	0000000A		
JPI\$-PRCNAM	=	0000031C		PGFLQUOTA	0000018A	R	02	
JPI\$-PRI	=	00000302		PGFLQUOTAL	0000018E	R	02	
JPI\$-PRIB	=	00000309		PHD\$Q_PRIVMSK	=	00000000		
JPI\$-PROCPRIV	=	00000204		PID	00000190	R	02	

SATSSS35
Symbol table

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) K 10
16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 34
(3)

PID1	0000023B	R	04	QUOTA_LIST	000001B6	R	03
PIDL	00000194	R	02	REG	0000006D	R	04
PID_STR	00000163	R	03	REGNUM	0000007F	R	04
PNS	00000109	R	03	REG_CHECK	00000BFD	R	05
PPGCNT	00000196	R	02	REG_SAVE	00000BF3	R	05
PPGCNTL	0000019A	R	02	REG_SAVE_AREA	00000008	R	04
PQLS_ASTLM	= 00000001			RETADR	0000005D	R	04
PQLS_BIOLM	= 00000002			RMS\$ FNF	*****	X	05
PQLS_BYTLM	= 00000003			SATSSS35	00000000	RG	05
PQLS_CPULM	= 00000004			SERV_NAME	0000021F	R	04
PQLS_DIOLM	= 00000005			SHORT_LIST	000001FA	R	03
PQLS_FILLM	= 00000006			SHR\$ ABEND	= 000010E0		
PQLS_LISTEND	= 00000000			SHR\$ BEGIND	= 00001038		
PQLS_PGFLQUOTA	= 00000007			SHR\$ ENDEDD	= 00001080		
PQLS_PRCLM	= 00000008			SHR\$ TEXT	= 00001130		
PQLS_TQELM	= 00000009			SS\$ NORMAL	*****	X	05
PQLS_WSDEFAULT	= 0000000B			STATE	000001C9	R	02
PQLS_WSQUOTA	= 0000000A			STATEL	000001CD	R	02
PRCCNT	0000019C	R	02	STATUS	00000065	R	04
PRCCNTL	000001A0	R	02	STEP	= 00000016		
PRCLM	00000030	R	02	STP0	0000003D	R	05
PRCLML	00000034	R	02	STP1	000000BE	R	05
PRCNAM	000001A2	R	02	STP10	0000049C	R	05
PRCNAML	000001B1	R	02	STP11	00000543	R	05
PRI	000001BD	R	02	STP12	000006CC	R	05
PRIB	000001C3	R	02	STP13	0000075E	R	05
PRIBL	000001C7	R	02	STP14	000007FF	R	05
PRIL	000001C1	R	02	STP15	00000824	R	05
PRINT_FAIL	00000C3F	R	05	STP16	000008CF	R	05
PRIVMASK	00000051	R	04	STP17	00000962	R	05
PRIVS	000002B3	R	04	STP18	0000099E	R	05
PRIV_ARGS	= 00000002			STP19	00000A0A	R	05
PROCPRIV	000001B3	R	02	STP2	000000F3	R	05
PROCPRIVL	000001BB	R	02	STP20	00000A4A	R	05
PROC_NAME	0000054E	R	03	STP21	00000AC1	R	05
PROC_UIC	00000565	R	03	STP22	00000B48	R	05
PRVSV_CMEXEC	= 00000001			STP3	00000146	R	05
PRVSV_CMKRN	= 00000000			STP4	0000018D	R	05
PRVSV_DETACH	= 00000005			STP5	000001E4	R	05
PRVSV_DIAGNOSE	= 00000006			STP6	00000224	R	05
PRVSV_GROUP	= 00000008			STP7	0000027F	R	05
PRVSV_GRPNAM	= 00000003			STP8	00000309	R	05
PRVSV_LOG_IO	= 00000007			STP9	000003AA	R	05
PRVSV_NETMBX	= 00000014			STS	000001CF	R	02
PRVSV_NOACNT	= 00000009			STSV_FAC NO	= 00000010		
PRVSV_PHY_IO	= 00000016			STSV_INHIB_MSG	= 0000001C		
PRVSV_PRCB	= 0000000A			STSL	00000134	R	03
PRVSV_PRCMBX	= 0000000B			SUCCESS	000001D3	R	02
PRVSV_PSWAPM	= 0000000C			SYSSASCEFC	= 00000001		
PRVSV_SETPRI	= 0000000D			SYSSCMKRN	*****	GX	05
PRVSV_SETPRV	= 0000000E			SYSSCREMBX	*****	GX	05
PRVSV_SYSNAM	= 00000002			SYSSCREPRC	*****	GX	05
PRVSV_SYSPRV	= 0000001C			SYSSDACEFC	*****	GX	05
PRVSV_TMPMBX	= 0000000F			SYSSDELPRC	*****	GX	05
PRVSV_VOLPRO	= 00000015			SYSEXIT	*****	GX	05
PRVMASK	00000196	R	03	SYSSFAO	*****	X	05
PRVPRT	00000050	R	04				

SATSSS35
Symbol table

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) L 10
16-SEP-1984 00:50:17 VAX/VMS Macro V04-00
5-SEP-1984 04:30:34 [UETPSY.SRC]SATSSS35.MAR;1

Page 35
(3)

SYSS\$FAOL	*****	GX	05
SYSS\$GETCHN	*****	GX	05
SYSS\$GETJPI	*****	GX	05
SYSS\$HIBER	*****	GX	05
SYSS\$PUTMSG	*****	GX	05
SYSS\$QIOW	*****	GX	05
SYSS\$SETAST	*****	GX	05
SYSS\$SETPRN	*****	GX	05
SYSS\$SETPRV	*****	GX	05
SYSS\$WAITFR	*****	GX	05
SYSS\$WAKE	*****	GX	05
TEST_MOD_BEGIN	00000019	R	03
TEST_MOD_FAIL	0000002A	R	03
TEST_MOD_NAME	00000000	R	03
TEST_MOD_NAME_D	00000009	R	03
TEST_MOD_SUCC	0000001F	R	03
TEST_PID	000002AF	R	04
TMBU	000001D5	R	02
TMBUL	000001D9	R	02
TMD_ADDR	0000004C	R	04
TMN_ADDR	00000048	R	04
TPID	00000000	R	04
TQCNT	000001DB	R	02
TQCNTL	000001DF	R	02
TQLM	00000036	R	02
TQLML	0000003A	R	02
UETP	= 00740000		
UETPS_ABEND	= 007410E0		
UETPS_BEGIN	= 00741038		
UETPS_ENDED	= 00741080		
UETPS_SATSMS	= 007480D9		
UETPS_TEXT	= 00741130		
UIC	0000003C	R	02
UICL	00000040	R	02
UIC_MSG	00000144	R	03
UM	0000017C	R	03
USERNAME	00000042	R	02
USERNAMEL	0000004E	R	02
VIRTPEAK	000001E7	R	02
VIRTPEAKL	000001EB	R	02
VOLUMES	000001E1	R	02
VOLUMESL	000001E5	R	02
WSAUTH	000001ED	R	02
WSAUTHL	000001F1	R	02
WSPEAK	000001F9	R	02
WSPEAKL	000001FD	R	02
WSQUOTA	000001F3	R	02
WSQUOTAL	000001F7	R	02
WSSIZE	000001FF	R	02
WSSIZEL	00000203	R	02

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
ITEM_LIST	00000205 (517.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
RODATA	00000569 (1385.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	000002BB (699.)	04 (4.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSSS35	00000E3B (3643.)	05 (5.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.08	00:00:00.31
Command processing	134	00:00:00.63	00:00:02.15
Pass 1	472	00:00:19.08	00:00:32.37
Symbol table sort	0	00:00:01.51	00:00:01.52
Pass 2	257	00:00:04.93	00:00:08.65
Symbol table output	46	00:00:00.38	00:00:00.54
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	949	00:00:26.66	00:00:45.59

The working set limit was 2000 pages.
116238 bytes (228 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1010 non-local and 25 local symbols.
1209 source lines were read in Pass 1, producing 42 object records in Pass 2.
63 pages of virtual memory were used to define 57 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SHRLIB]UETP.MLB;1	10
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	2
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	41
TOTALS (all libraries)	53

1207 GETS were required to define 53 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SATSSS35/OBJ=OBJ\$:SATSSS35 MSRC\$:SATSSS35/UPDATE=(ENH\$:SATSSS35)+EXECML\$/LIB+SHRLIB\$:UETP/LIB

0422 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY